

**Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology**



**(Deemed to be University Estd. u/s 3 of UGC Act, 1956)**



**School of Computing**

**B.Tech. – Computer Science and Engineering**  
**VTR UGE2021- (CBCS)**

Academic Year: 2025–2026

SUMMER SEMESTER - SS2526

Course Code : 10211CS207

Course Name : Database Management Systems

Slot No : S1L4

# DBMS TASK - 7 REPORT

**Title: Triggers, Views and Exceptions**

Submitted by:

VTUNO	REGISTER NUMBER	STUDENT NAME
VTU28992	24UECS0793	O. SAI MANIKANTA REDDY

## TASK7:

### Triggers, Views and Exceptions

#### Aim:

To Conduct events, views, exceptions on CRUD operations for restricting phenomenon.

a) To create a trigger in PL/SQL that automatically inserts a new record in the match\_result table when a new record is inserted into the match table.

b) To create a view that displays the details of players along with their team details.

c) To write a non-recursive PL/SQL procedure to retrieve even-numbered PlayerIDs registered for any Match with exception handling.

- **To create a trigger in PL/SQL that automatically inserts a new record in the match\_result table when a new record is inserted into the match table.**

```
SQL>CREATE TABLE Match_Result (ResultIDvarchar(10) PRIMARY KEY,  
MatchIDvarchar(10), TeamIDvarchar(6), Result varchar(20), FOREIGN KEY(MatchID)  
REFERENCES Match(MatchID), FOREIGN KEY(TeamID) REFERENCES Team(TeamID));
```

```
SQL>CREATE OR REPLACE TRIGGER trg_insert_match_result AFTER INSERT ON Match  
FOR EACH ROW
```

```
DECLARE
```

```
N NUMBER;
```

```
BEGIN
```

```
N:=INSTR(UPPER(:NEW.RESULT),'WIN');
```

```
IF N>0 AND INSTR(UPPER(:NEW.RESULT),UPPER(:NEW.TEAMID1))>0 THEN
```

```
INSERT INTO Match_Result(ResultID, MatchID, TeamID, Result)
```

```
VALUES ('R'||:NEW.MatchID||'1', :NEW.MatchID, :NEW.TeamID1, 'WIN');
```

```
INSERT INTO Match_Result(ResultID, MatchID, TeamID, Result)
```

```
VALUES ('R'||:NEW.MatchID||'2', :NEW.MatchID, :NEW.TeamID2, 'FAIL');
```

```
ELSE
```

```
INSERT INTO Match_Result(ResultID, MatchID, TeamID, Result)
```

```
VALUES ('R'||:NEW.MatchID||'1', :NEW.MatchID, :NEW.TeamID1, 'FAIL');
```

```
INSERT INTO Match_Result(ResultID, MatchID, TeamID, Result)
```

```
VALUES ('R'||:NEW.MatchID||'2', :NEW.MatchID, :NEW.TeamID2, 'WIN');
```

```
END IF;
```

```
END;
```

```
SQL>insert into matchvalues('M01','CCB01','TCB01','22-JUN-2022',1.3,'TCB01 - WIN');
```

```
SQL>insert into matchvalues('M02','CCB02','TCB02','22-JUN-2022',8.3,'CCB01 - WIN');
```

```
SQL>insert into matchvalues('M03','TRIB01','TCB01','24-JUN-2022',8.3,'TCB01 - WIN');
SQL>insert into matchvalues('M04','TRIB01','TCB02','25-JUN-2022',8.3,'TRICB01 - WIN');
SQL>insert into match values('M05','TCB01','TCB02',DATE '2025-09-20',16.00,'TCB02 - WIN');
```

```
SQL> Select * from Match_Result;
```

**Write the Output**

- **To create a view that displays the details of players along with their team details.**

```
SQL>CREATE OR REPLACE VIEW Player_Team_View AS SELECT p.PlayerID, p.FName
AS FirstName, p.PlayingRole AS PlayingRole, p.Batting AS BattingStyle, p.Bowling AS
BowlingStyle, t.TeamID AS TeamID, t.Name AS TeamName, t.Coach AS Coach, t.Captain AS
Captain FROM Player p JOIN Team t ON p.TeamID = t.TeamID;
```

```
SQL> Select * from Player_Team_View;
```

**Write the Output**

- **To write a non-recursive PL/SQL procedure to retrieve even-numbered PlayerIDs registered for any Matchwith exception handling.**

```
CREATE OR REPLACE PROCEDURE Get_Even_PlayerIDs_For_Matches IS
CURSOR even_players IS
SELECT DISTINCT p.PlayerID, p.FName, p.LName, m.MatchID
FROM Player p
JOIN Team t ON p.TeamID = t.TeamID
JOIN Match m ON (m.TeamID1 = t.TeamID OR m.TeamID2 = t.TeamID)
WHERE MOD(TO_NUMBER(REGEXP_SUBSTR(p.PlayerID, '\d+$')), 2) = 0;
```

```
v_countNUMBER := 0;
```

```
BEGIN
```

```
DBMS_OUTPUT.PUT_LINE('Even-numbered PlayerIDs registered for Matches:');
```

```
FOR rec IN even_players LOOP
```

```
v_count := v_count + 1;
```

```
DBMS_OUTPUT.PUT_LINE('MatchID: ' || rec.MatchID ||
```

```
' | PlayerID: ' || rec.PlayerID ||
```

```
' | Name: ' || rec.FName || ' ' || rec.LName);
```

```
END LOOP;
```

```
-- Handle case where no records found
IF v_count = 0 THEN
    RAISE NO_DATA_FOUND;
END IF;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No even-numbered PlayerIDs found for any match.');
```

When OTHERS THEN

```
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
/
```

SQL> set serveroutput on

SQL> /

Procedure created.

SQL>SET SERVEROUTPUT ON;

SQL>EXEC Get\_Even\_PlayerIDs\_For\_Matches;

**Write the Output**

**Result:**

Thus the Triggers, Views and Exceptions experiment was successfully completed results are verified.